XEROX
Office Products Division
Systems Development Department

To: Distribution

From: Ron Crane

Subject: **Dandelion Display**

Date: February 13, 1980

Org: SDD/SDT

Filed: [IRIS]<Workstation>HSIO>Doc>WSDisp.press

Dist.: Workstation Design Group:

Overview

This memo describes the Dandelion display controller and associated clock generation logic, all located on the high speed I/O board. This document covers what functions are performed by the display controller/microcode, how they are partitioned between hardware and microcode, how the microcode controls the hardware, and finally how the hardware functions are implemented.

Display Functions

Dandelion supports a 17" raster scan display with an active image of 798 lines by 1024 bits per line (using 51,072 words of memory) and a border area of 32 lines at the top and bottom of the screen and 32 bits on each side of the screen. The total display is 897 lines x 1088 bits. The screen frame rate is 38.7 frames/sec. A 16x16 addressable cursor is supported as well as the capability to scroll subwindows. The display controller takes words from a bitmap in the low 64K bank of the memory system and produces video and sync signals for a 17" monitor. The border consists of a repeated pattern coming from a border pattern register register. Memory refresh is also performed by the display microcode.

The display controller uses a partitioned, two-port memory to reduce the loss of processor bandwidth while the display is running. The display blocks processor access to the low 64K memory bank only when it is shifting out data bits during an active horizontal line. The procesor has complete access to the low bank at all other times (i.e. during inactive lines, retrace, and blanking intervals). When the picture is turned off, the low memory bank is identical in performance to the high banks. The display cannot access the higher banks of memory and has no effect on processor access to these banks.

The following functions are performed by the display controller hardware/microcode.

1. Read data from memory and shift out blocks of 1024 bits.
2. Provide horizontal sync, vertical sync, and blanking signals.
3. Insertion of 16 x 16 cursor on the display.
4. Perform memory refresh.

Partitioning Functions between Hardware and Microcode

The tasks required of the display controller are hierarchical in time and span a wide range of times (shifting bits, reading words, blanking, horizontal sync, vertical sync, fields, & frames). It is important to minimize the amount of hardware used for any individual controller and at the same time, not use an excessive amount of the processor for a single I/O function. For the display controller, a horizontal line period (28.8 uS) was chosen as the dividing point between functions implemented in hardware and microcode. Memory accesses, parallel to serial conversion, and horizontal sync generation are done in hardware. Line counting, vertical sync, cursor insertion, and memory refresh are handled with microcode.

Microcode - Hardware Interface

Three registers comprise the microcode-hardware interface. They are described below and summarized in the following figure. Use of this interface to operate a display will be described in the next section. The following words will appear in this and the following section.

Line Segment - A part or all of a horizontal line in which the displayed words come from contiguous memory locations. A line segment can be between 1 and 64 words long. The sum of the line segments which comprise a horizontal line is always 64 words. Each entry in the control fifo described below specifies one line segment.

Window - A rectangular region on the display made up of line segments on successive scan lines. The line segments all start and end on the same word boundaries, i.e. they are all placed one above another.

Cursor - This is a special case of window which is 16 scan lines high and two words wide. Contained in this region is a 16x16 array which is bit aligned. The remaining area in the two word wide area not covered by the 16x16 array is typically loaded with those bits from the main bit map which would be visible if the cursor window were not present.

Control Register

This register contains 7 bits which control the display operation.

On - This bit enables requests to the processor for service during the display click. These requests occur once per horizontal line. This bit does not affect memory accesses.

Blank (Bk) - Setting this bit causes the entire horizontal line to be blanked. No memory accesses will occur when set, independent of other bit settings. Typically, blank will be set during vertical refresh.

Picture (Pic) - Setting this bit will cause memory accesses if blank is not set. The contents of the control fifo is used to specify which locations are accessed and displayed. If both Pic and Bk are cleared, the border pattern will be repeated for all bits within a line and no memory accesses will occur.

Invert (Inv) - Setting this bit causes inversion of the entire visible portion of the screen (main picture and border).

Odd (OD) - Setting this bit indicates to the controller that the odd field of a frame is being scanned. This is used by the controller to determine whether vertical sync pulse should start and stop at the beginning or middle of a line. It starts at the middle for an odd line. This bit should not be changed during a vertical sync pulse, since changing it during the sync pulse would cause the end of the sync pulse to occur at a different location in a line from where it started. Neglecting this could cause interlace problems on monitors which triggered on the trailing edge of vertical sync. (Most of our monitors are triggered on the leading edge of vertical sync.)

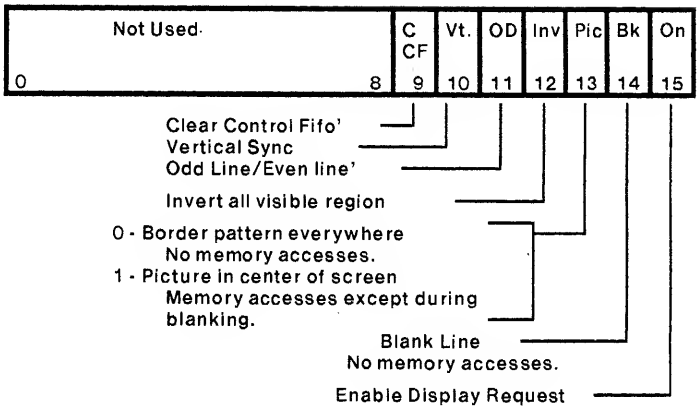
Vertical (Vt.) - Vertical sync pulse line goes low when this bit is set. The time of the transition relative to a horizontal line time is determined by the odd bit.

Clear Control Fifo' (CCF) - When set to zero, this bit causes the control fifo to be cleared. Normally this bit is kept set to one.

Dandelion Display Controller Registers

Control Register (on X Bus)

DCtl in any cycle.



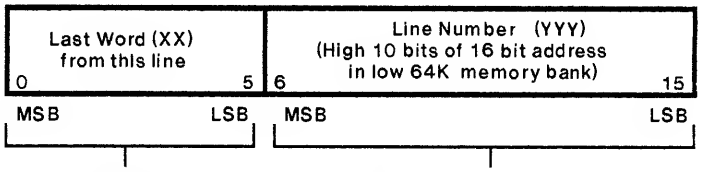
Register controls display operation.

It is cleared to 0 by IOPReset when system is powered on. When cleared, the display will receive only horizontal sync, video will be the contents of border register at power-up, and there will be no display requests to the processor.

Vertical sync is a strobed version of the vertical bit in the control register. To produce interlaced scan, vertical sync is strobed and changes at the beginning of the line for even lines and middle of the line for odd lines, as specified by bit 11.

Control Fifo Register (on Y Bus)

DCtlFifo← in any cycle



One word is written into Fifo for each cycle in which DCtlFifo is asserted. Used to specify a location (line segment) in memory from which to retrieve data for display. The low 10 bits specifies the line #, and the high 6 bits specify the last location to be read from that line. If the last location is not the end of a line (= 63) then the next Fifo entry is used to identify the next line # from which a group of words will be taken. Note that the low 6 bits (word #) used for the address counts from 0 to 63 for each line. The control fifo only permits selecting the line number and the location along the line at which a transition is made from one line to another. Thus, as viewed on the monitor screen, this mechanism facilitates vertical movement of images, but not horizontal movement, since low 6 bits of address come from word counter. One control word is loaded into the fifo for each continuous segment of words in a horizontal line. Thus, a normal line with no cursor or window will have one control word. A line with a cursor in the middle will have 3 control words. While the fifo size is 16 words, no more than 10 entries should be for a single line. The last control word for a line should specify word 63 (decimal). The controller will "wrap around" to the next scan line in a field if necessary to advance to the word number specified in a control fifo entry. Control words can be loaded into the Fifo any time before the line in which they are used. Care must be taken not to insert extra control words, and lose sync. Clearing control fifo during vertical sync will purge the fifo. The control Fifo is cleared by setting bit 9 of control register to 0.

Selects location (word #) along horizontal line after which display will jump to a new line in memory from which it reads data.

Horizontal line starts reading data at

YYY	00

in main memory,

and continues reading until location

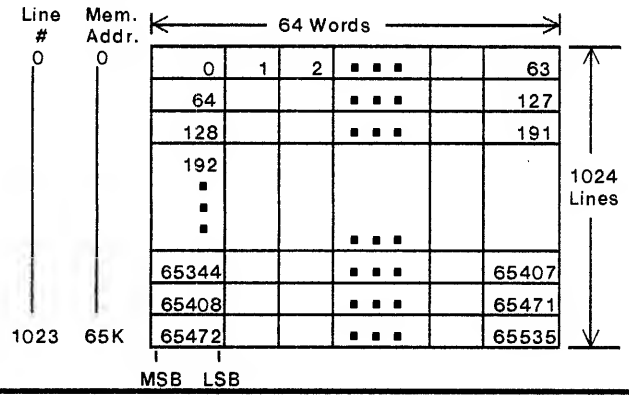
YYY	XX

AFTER which it advances to the next control Fifo location specifying the next line segment:

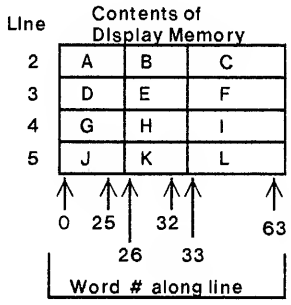
YYY'	XX + 1

on the same output line.

Display Memory (Low 64K bank of physical address space)



Example of Control Fifo Use



Contents of control fifo just before beginning of line X.

25	4
32	2
63	5

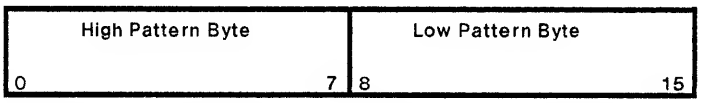
Line X as displayed on screen.

*	G	B	L	*

* 32 Bits of border pattern 1 byte repeated 4 times.

Border Pattern Register (on Y Bus)

DBorder← In any cycle



High pattern is repeated on lines 4n + 2, 4n + 3 where n is an integer.

Low pattern is repeated on lines 4n, 4n + 1 where n is an integer.

Border pattern is repeated on every line during the first and last 32 bits scanned. The high and low patterns are used on alternate pairs of lines. Lines are numbered starting from 0 at the top of the screen. The border pattern will be repeated all across a horizontal line if bit 13 of the control register is zero.

Control Fifo Register

This register contains two fields: *last word* and *line number* which are used to specify a line segment.

Last word is used to specify the number of the last word position (relative to lines aligned on 64 word boundaries in memory), to be used for a given line segment. *Last word* is the high 6 bits of the register, and typically remains constant for a given window.

Line number specifies the line number in memory from which the current line segment is taken. The low 10 bits of the register are the line number and are typically incremented by 2 for successive lines within a window.

Border Pattern Register

This register contains the two border pattern bytes. Each byte is repeated 4 times at the beginning and end of each horizontal line. The low border byte is used during lines 4n, and 4n+1 (lines 0,1,4,5,8,9...) and the high border byte is used during lines 4n+2 and 4n+3 (lines 2,3,5,6,10,11....). The border pattern register need be loaded only once.

Using the Controller

This section outlines the actions the microcode must take to get a bitmap from the low 64K of memory onto the display. The following figure shows what gets loaded into each register during the various parts of a frame. Note that the only difference between the fields in a frame are the setting of the odd field bit in the control register and the line offset used when loading the control fifo.

Register Loading Sequence to Get Bit Map on Display

# Lines	Function	Control Register Loaded only once per function during first line	Control Fifo Loaded once per line in even and odd fields.
2	End Vert Sync, Blk	3	——
16	Top Border	41_{16}	——
399	Even Field	45_{16}	Last Word = 63 Line Number = even #'s
16	Bottom Border	41_{16}	——
2	Start Blanking	43_{16}	——
14	Start Odd V Sync	73_{16}	——
2	End Vert Sync, Blk	3	——
16	Top Border	41_{16}	——
399	Odd Field	45_{16}	Last Word = 63 Line Number = odd #'s
16	Bottom Border	41_{16}	——
2	Start Blanking	43_{16}	——
13	Start Even V Sync	63_{16}	——

897 lines total 448.5 lines/field

To add a cursor, the control fifo is loaded with 2 or 3 segments per line for a run of 8 lines in each field. This is shown in the next figure. A window in addition to the cursor simply adds more segments per line. For both the cursor and the window, some computation must be made once per frame to determine the control fifo entries, and for the cursor, there must also be an updating of the cursor bitmap each time the cursor moves.

Register Loading Sequence to Get Bit Map with Cursor

# Lines	Function	Control Register Loaded only once per function during first line	Control Fifo Loaded once per line in even and odd fields.
2	End Vert Sync, Blk	3	——
16	Top Border	41_{16}	——
x	Even Field	45_{16}	Last Word = 63 Line Number = even #'s
8	Even Cursor	45_{16}	3 Seg. for cursor
391-x	Even Field	45_{16}	Last Word = 63 Line Number = even #'s
16	Bottom Border	41_{16}	——
2	Start Blanking	43_{16}	——
14	Start Odd V Sync	73_{16}	——
2	End Vert Sync, Blk	3	——
16	Top Border	41_{16}	——
y	Odd Field	45_{16}	Last Word = 63 Line Number = odd #'s
8	Odd Cursor	45_{16}	3 Seg. for cursor
391-y	Odd Field	45_{16}	Last Word = 63 Line Number = odd #'s
16	Bottom Border	41_{16}	——
2	Start Blanking	43_{16}	——
13	Start Even V Sync	63_{16}	——

897 lines total 448.5 lines/field

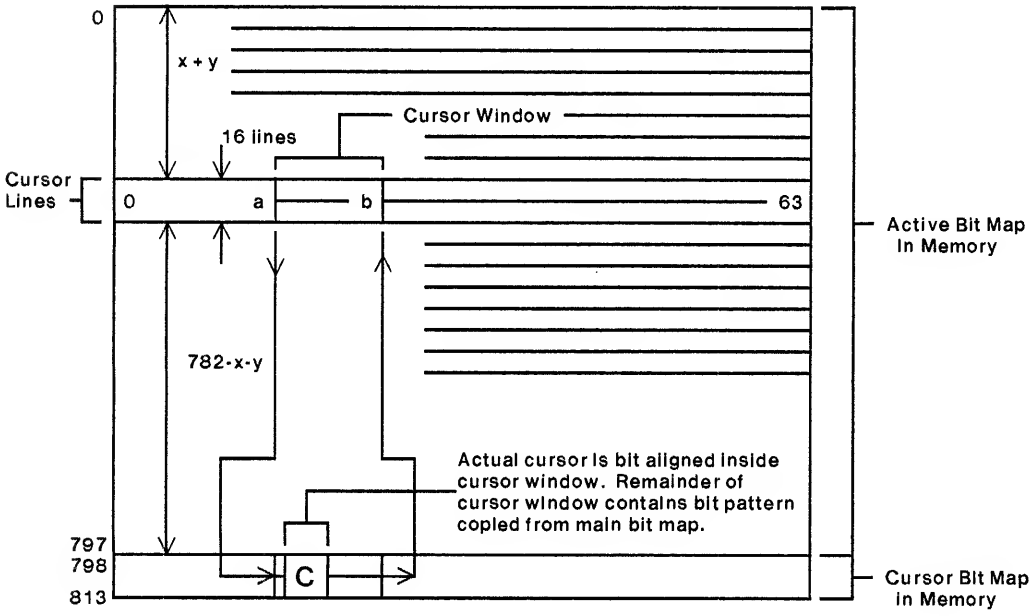
During Cursor, 3 control fifo entries per line are used.

Last Word	Line Number
a	x + y + 2n
b	798 + 2n
63	x + y + 2n

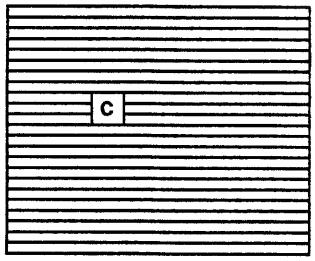
a	x + y + 1 + 2n
b	799 + 2n
63	x + y + 1 + 2n

n is line increment within cursor ranging from 0 to 7 for successive lines in a field.

Memory



Display Screen



During Cursor lines, each line is divided into 3 segments. The middle segment comes from the cursor bitmap.

Segment 1 Word 0 to a
Segment 2 Word a + 1 to b
Segment 3 Word b + 1 to 63

Hardware Implementation

Display Controller (Horizontal line generator)

The display hardware handles only those functions which repeat on a horizontal line basis. The time of a line is small enough that the dedicated hardware to handle it is well utilized. Similar hardware for counting lines and doing vertical tasks does not provide a similar payoff, because these things do not happen often enough to use much of the processor.

Horizontal Events

A horizontal line starts with 32 bits of border pattern, then 1024 bits of picture, 32 bits of right border, and 382 bits of blanking. A horizontal sync pulse starts 8 bits after blanking starts and ends 8 bits before blanking ends.

Vertical Events

A frame consists of an even and an odd field, each of which contains 16 lines of top border pattern, 399 lines of picture, 16 lines of bottom border, and 17.5 (17 lines in one field & 18 lines in other) lines of blanking during which vertical retrace takes place. The section covering controller use further describes vertical events. No further mention of vertical events will appear in this section.

Display Controller Logic

The next two figures show a functional block diagram of the display controller and output machine timing diagrams. It has three principal parts; the output machine, a data fifo, and the read machine. Associated with the output machine is the control register. The border pattern register is associated with the data fifo. The read machine contains the control fifo and associated control fifo register, end condition logic to terminate memory accesses at the end of a round and at the end of a line, and the LRAS and LCAS memory clock generation. Following are descriptions of these sections.

Output Machine and Control Register

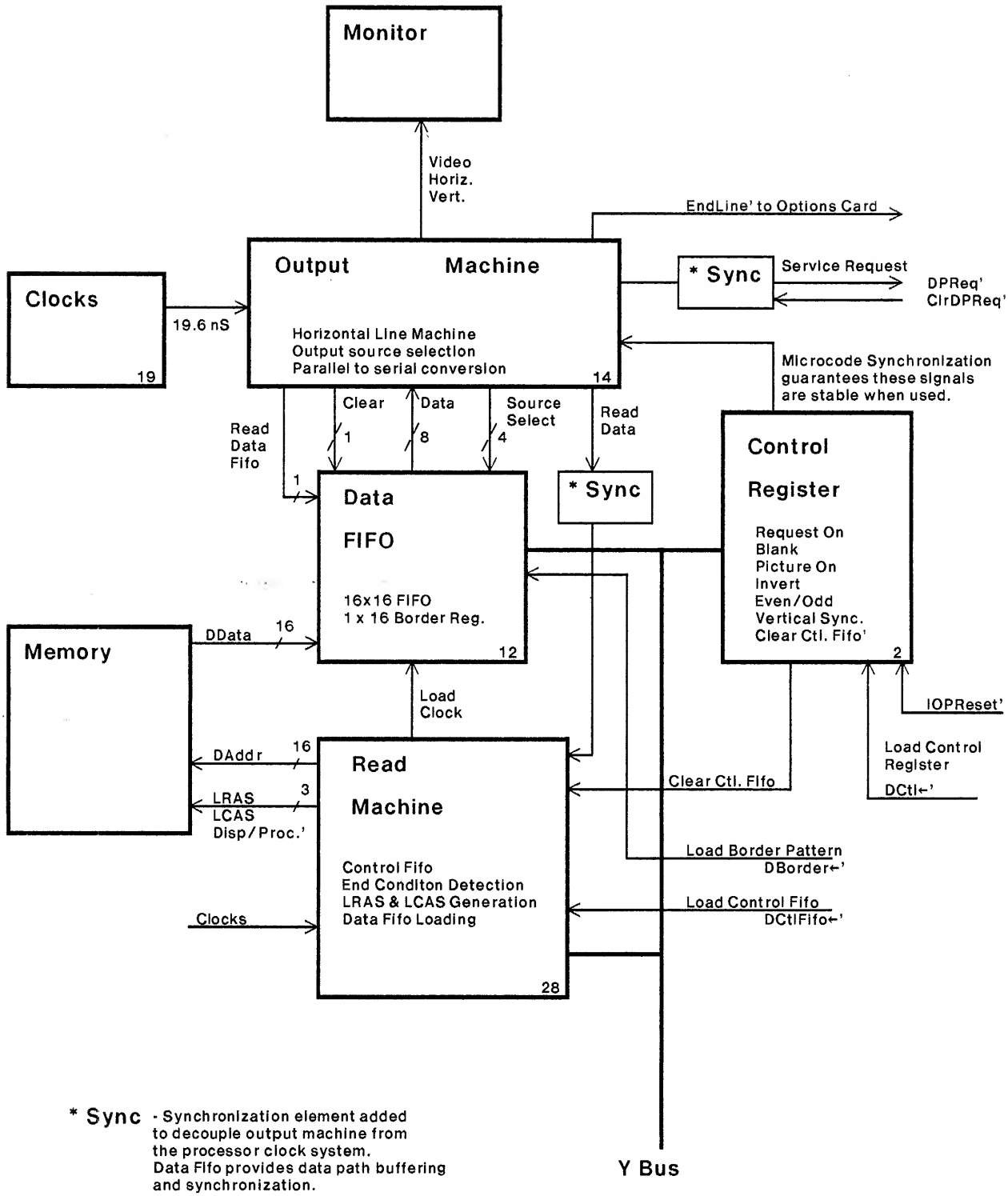
The output machine will be described in terms of the actions that take place during the output of a horizontal line. Each horizontal line starts with 32 bits of border pattern, followed by 1024 bits of data from memory, followed by 32 bits of border pattern, and ending with a blanking period during which the horizontal retrace takes place. This sequence repeats every horizontal line and is shown in the output machine timing diagram.

The output machine is controlled by a prom state machine with 210 states (1 state per machine cycle). It is cycled through these states by the display prom counter, which is a part of the system clock. The timing diagram shows the outputs of the prom register marked with asterisks. There are two time references in this figure. There are 1470 bits per line and they are marked on the top of the figure. There are 210 cycles of 7 bit times each, which are labeled in italics.

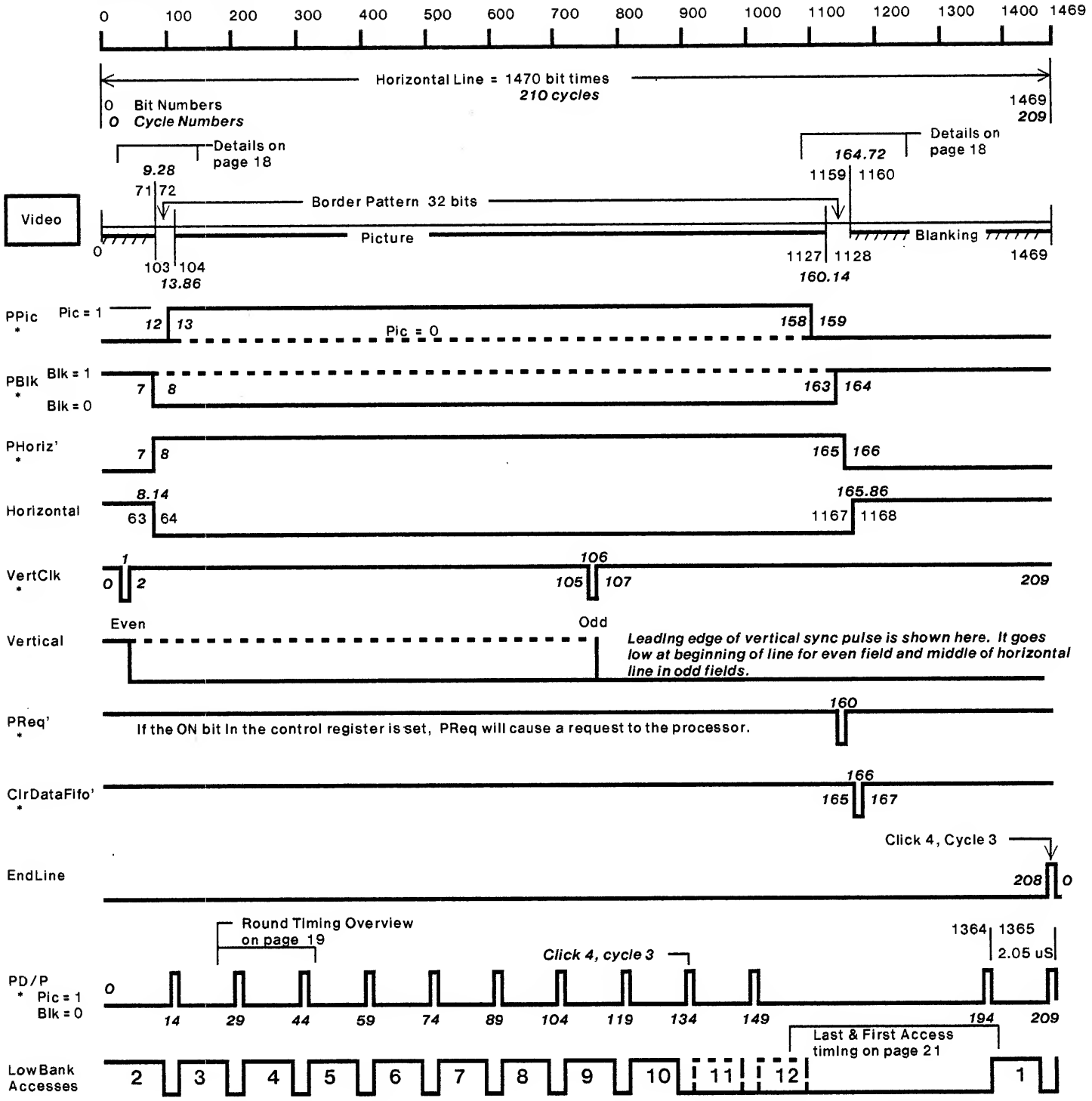
Starting at bit position 0 (look at line labeled video), the F16 counter in the output machine has just been resynchronized to 0 by the EndLine and Tick7' pulse. The output shift register and blanking register are strobed at the beginning of bit 8 and every 8th bit thereafter during a horizontal line. Thus, shift register loading, blanking, and unblanking are done on byte intervals.

The first 32 bits of a line come from the border pattern register. The selection of the high or low byte is done by a flip-flop which is toggled every horizontal line in a field. This produces the signal BPBS (border pattern byte select). This signal is always reset by a vertical sync pulse so the first line of a field comes from the low byte of the border pattern register. (More specifically, it is the first line after the trailing edge of the vertical sync pulse. Note that since the first few lines of a field are often blanked, it may not correspond to the first visible line.) Selection of the border pattern register outputs instead of the data fifo outputs is done by a sampled version of the the signal PPic from the prom register. It is

Display System Functional Block Diagram



*** Sync** - Synchronization element added to decouple output machine from the processor clock system. Data Fifo provides data path buffering and synchronization.



There are low bank accesses during rounds 1-10 for a line with 1 segment. Some amount of round 11 is used for lines with 2-10 segments. While not recommended, round 12 is available for lines with more than 10 segments. Care must be taken here since the data fifo can go empty in this round if more than 10 segments are specified. Word must be strobed into DataFifo at least 330 ns before it is read out. At beginning of line, DataFifo is filled to 13 words at the end of round 2.

Note: Bit numbering starts at bit 0 and ends in 1469.
Cycle numbering starts with cycle 0 and is shown in italics. Cycle and bit numbering are referenced to the signal EndLine which goes high in the last cycle (209) of the Prom Counter before it resets. PromCounter has outputs DProm 0-7 which cycle from 46-255. These constitute 8 bits of address into the prom whose outputs are strobed into the prom register. To obtain the prom definition, 45 must be added to the above numbers for the address input.

* These signals are output of prom register.
PBlk, PPic, & PHoriz' are sampled every 8 bit times starting at the end of bit 7. Any transition at the output of the prom register must occur 2 or more bit times before sampling. The horizontal sync pulse comes directly from the synchronizing register. PPic and PBlk are delayed 1 sample interval (8 bit times) from synchronizing register before they take effect.

3 cycles/ click	210 cycles/ line	Bit Time = 19.59 nS	Cycle Time = 137.14 nS
5 clicks/ round	7 bits/ cycle	Bit clock = 51.04 MHz	Round Time = 2.0572 uS
14 rounds/line	1470 Bit times/line		Line Time = 28.8 uS

Timing: Output Machine

sampled every 8th bit time just after a byte has been read. Changes in PPic and PBlk must occur two bit times before they are sampled. (This means that the low-high edge of the 51 MHz clock, that eventually causes clocking of the prom register, must occur 2 bit times before the 51 MHz clock edge that causes the sampling to occur.)

On the first cycle boundary after the 4th border byte is loaded, PPic goes to a logic 1, such that the next byte loaded comes from the high byte of the data fifo. Byte selection is performed by the high bit of the bit counter. The fifo is clocked after the high byte is loaded. This process continues until all 64 words have been loaded into the shift register and shifted out. While the low byte of the 64th word is being shifted out, PPic goes low so that the next byte to go out comes from the border register. While the 4th border byte is being shifted out, PBlk comes on so that blanking starts on the next byte boundary. Blanking continues until the end of bit 71 (after the counter wraps around), after which the next horizontal line starts with the border pattern again.

The horizontal sync pulse starts 8 bit times after blanking starts and ends 8 bit times before blanking ends. Both horizontal and vertical sync signals pass through a low pass filter which increase the rise and fall times to approximately 100 nS. This helps reduce high-frequency radiation from the cable going to the monitor.

Data Fifo

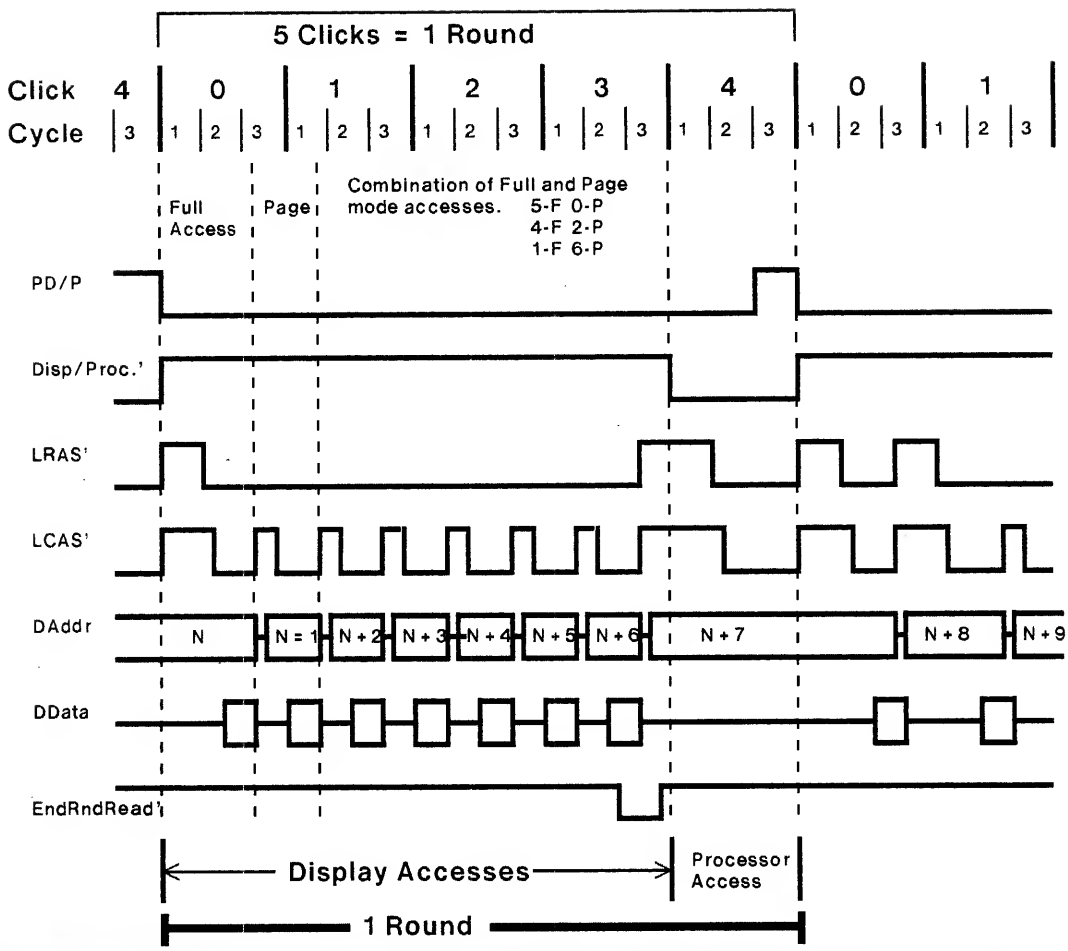
A 16 word data fifo provides buffering and solves the problem of synchronization between the memory system and the output machine. (While both memory and output machine run from the same clock, the largest common period is the 19.6 nS clock period which is too fine to be of any value.) Data is strobed into the holding register and fifo with DCAS' and DCASDly', respectively, both of which come from the read machine. Words are read out of the fifo with the signal ReadDataFifo, which comes from the output machine. The border pattern register is loaded from the Y-Bus. The outputs of both the data fifo and border register are multiplexed onto a byte wide tri-state bus, then through TTL-ECL converter to the parallel input of the output shift register in the output machine. Selection of the appropriate output byte is done by the output machine. The data fifo is implemented with 4 74S225 16x5 fifo chips. The output machine controls the read machine such that the fifo never overflows or underflows during a line. Consequently, the input ready and output ready signals from the fifo need not be used.

Read Machine

The read machine does memory accesses during the first 4 clicks of a round. It always starts at the beginning of a round and continues to either the end of the round or the last word of a line has been accessed, whichever occurs first. Reads in a round are initiated by a signal, PD/P, from the output machine. The read machine will determine the mix of full and page mode accesses necessary and do the maximum number of memory accesses possible within a round. The low 6 bits of the memory address always count from 0 to 63. The high 10 bits (line number) are specified in the control fifo entry. The last word to be used from a given line is also specified in the control fifo entry (6 bits) and is used to advance to the next fifo entry when that word number is reached. The three parts of the read machine (control fifo, end condition logic, and LRAS, LCAS generation) are described in the following paragraphs.

Control Fifo

The control fifo contains 16 entries. Each entry identifies a line segment using 10 bits to specify the line number and 6 bits to specify the last word in the segment. The control fifo is loaded from the Y-Bus, unloaded by a signal from the end condition logic, and cleared by a bit from the control register. The control fifo uses 74S225 16x5 fifo chips. Neither the input ready or output ready status signals are used. The microcode must take care to load only the entries for one scan line per horizontal line wakeup on the average. The control fifo should be cleared once per vertical field to eliminate the effects of noise and assure its state at the beginning of a field.



General

Full Access = 293 nS
Page Access = 215 nS

The beginning of display accesses synchronized with the beginning of click 1 and run asynchronously with respect to the processor. Once started, however, they always complete within 4 clicks.

Number of accesses in Round

The number of accesses in a round is determined by the combination of page & full accesses. Full accesses occur on line segment boundaries, the first access in a round, and every 8th access. The EndCount PROM uses this information to determine how many accesses can fit into the 4 display access clicks of a round. See page 20 of logic dwg.

Data Fifo Timing Details

It takes some time for data to percolate through the data fifo. The fifo also has finite size. This affects the relationship of the beginning and end of display accesses to begin and end of video bit stream.

LRAS & LCAS Generation

LRAS & LCAS are different for full and page mode display accesses and processor accesses. Timing for the state machine that generates these signals is on page 22 of logic dwg.

Read Machine: Round Timing Overview

Word Counter & End Condition Logic

The word counter counts from 0 to 63, synchronous with the memory accesses used to fill the data fifo. The output of this counter is compared with the 6 bit last word field of the current control fifo entry. When they are equal, the control fifo is advanced to the next entry. There is also logic to determine when a full (RAS and CAS) memory reference should take place. A full reference must take place whenever one of the RAS bits at the memory chips changes. This can occur on the first reference in a round, when the control fifo is advanced, and on every 8th memory reference due to the arrangement of bits in the memory system. (This seemingly odd arrangement of bits in the memory system was necessary due to the late arrival of bits 8-11 at the processor port for memory address. Bit 12 had to be used for a row address instead of bit 8.) Whenever one of these 3 conditions for full access occurs, the F/P' line to the LRAS-LCAS circuits goes high.

The number of accesses in a round depends on the number of full (293 nS) and page mode (215 nS) accesses that occur. A maximum of 5 full accesses, 4 full and 2 page accesses, or 1 full and 6 page accesses can occur. Thus the total number of accesses can range from 5 to 7. A prom state machine looks at the combination of accesses and drops the signal EndRndRead' during the last access of a round. The accesses in a round can end early if word 63 is reached. The signal InhibitRead also becomes true after word 63, locking out any further reads, independent of PD/P signal from output machine, until is reset by the signal ClrDataFifo' from the output machine. Details of the state machine and other logic timing are in the Clock and Display drawing package.

LRAS-LCAS Generation

The signals LRAS and LCAS are the clocks for the low bank of the memory system. These signals are identical to RAS and CAS for processor memory references (411 nS cycles), but have a faster full cycle time (293 nS) and a page mode cycle (215 nS) when the display is using the low bank (indicated by Disp/Proc' in the high state). In all cases, CAS follows RAS by 49 nS. A multiplexer preceeding the output registers is switched by the Disp/Proc' signal selecting either the main RAS-CAS generator or the display RAS-CAS generator. Both of these generators are simple state machines using one counter and discrete logic for decoding. They have a 19.6 nS cycle time. The CAS signal output register is clocked by an inverted version of the clock, such that the active edge of CAS is delayed 9.8 nS from the regular cycles of the state machine. Using both rising and falling clock edges is preferred to doubling the frequency of the clock for only a few signals.

Main RAS-CAS Generation

RAS' is logically similar to the signal Cycle1, it is high during cycle 1 and low during cycles 2 and 3. It must have a small propagation path from a common reference, however, and is generated in ECL by sampling and delaying the Cycle3' signal. CAS is derived from RAS by further delay. CAS is terminated at the same time as RAS by directly resetting the CAS output flip-flop with the ECLRAS' signal.

Display RAS-CAS Generation

Display RAS and CAS are generated as two separate signals, since CAS may have to cycle independently of RAS during page mode accesses. The generation logic uses a 4 bit ECL counter (F10016) and discrete logic elements to decode the RAS and CAS outputs. For full memory cycles, the counter is preset to 1 and counts to 15, resulting in 293 nS cycles. If a page mode cycle is to take place, the counter is preset with a 5, resulting in a 215 nS cycle. RAS and/or CAS precharge takes place at the beginning of the cycle. Memory data is latched and the word counter and end condition logic are strobed by DCAS' at the end of a cycle. The full/page signal (F/P') comes valid during a cycle and is used by the LRAS-LCAS state machine at the end of each cycle. This determines whether the counter is preset with 1 or 5 for a full or page cycle, or if round accesses are to terminate.

Round Access Termination - Round accesses are terminated by the signal EndRndRead' coming from the end condition logic. This signal is sampled at the end of the current cycle and initiates the termination sequence for display accesses. The state machine is preset and held in a state with both display RAS and CAS inactive. The signal Disp/Proc.' is brought low at the next click boundary.

Line Access Termination - Line accesses come to an end when the word counter reaches 63. This causes EndRndRead' to go low and subsequently the signal InhibitRead'. Round accesses are terminated, and in addition, InhibitRead' locks out further memory accesses by the display until it is reset by the signal ClrDataFifo' (which comes from the output machine).

Memory System Comments

The memory system has two ports; one to the processor and one to the display. In addition, it is partitioned such that the low 64K bank and display port can operate simultaneously with the processor port and all remaining banks of memory. All data paths are 16 bits. A common clock source (display) provides 137 nS and 411 nS ticks.

Memory must supply a word to the display 293 nS after the first request, and at 215 nS intervals thereafter until the end of the page (for page mode accesses), or the display decides it wants no more data.

Memory chips which meet the Xerox purchase specification for MK4116-2 parts will work in this system. The Xerox spec. is like the Mostek spec. except that the cycle time is 375 nS instead of 320 nS. The AC parameters for the chips are the same. In particular, CAS precharge = 60 nS, RAS precharge =100 nS min., RAS hold = 20 nS, CAS address setup = -10 nS, RAS access time = 150 nS, CAS access time = 100 nS. The system meets these AC parameters. Average cycle time is a thermal limitation. It is 375 nS min. for full accesses and 170 nS for page mode accesses. The period over which averaging takes place should not exceed the thermal time constant (milliseconds) of the memory chip. The averaging interval for the display is one horizontal line (28.8 uS). During this period, there are 64 display accesses (24 full and 40 page) and 26 possible processor (full) accesses. Using the minimum times, this would take only 25.55 uS. Thus the thermal limits are met.

The display port provides both regular and page mode access to the lower 64 K of memory only. The Display has priority for the lower 64 K. The processor port has access to all of memory. If the processor wants to access the low 64K, it must wait until the display is done. This is handled in the processor port by a wait line. If the high order bit of the address indicates the low 64K and the display is using it, the processor clock stops until the line goes low. Display timing is such that every 2 microseconds a 16 bit access is possible as well as during all blanking periods.

The partitioned memory approach relieves the processor of a significant and continuous drain of bandwidth without incurring a penalty in chip count. Performance numbers are shown in the bottom of figure 15 of logic and timing diagrams. The interface between the memory and display subsystems is fairly straightforward, thus providing modularity.

Processor and Memory Bandwidth

Approximately 9% of the processor clicks are consumed by display, cursor, and memory refresh tasks. Low 64K memory bank use is 60% due to display accesses and 10% due to display microcode, leaving 720K memory accesses/second (30% of its bandwidth) available to other tasks.